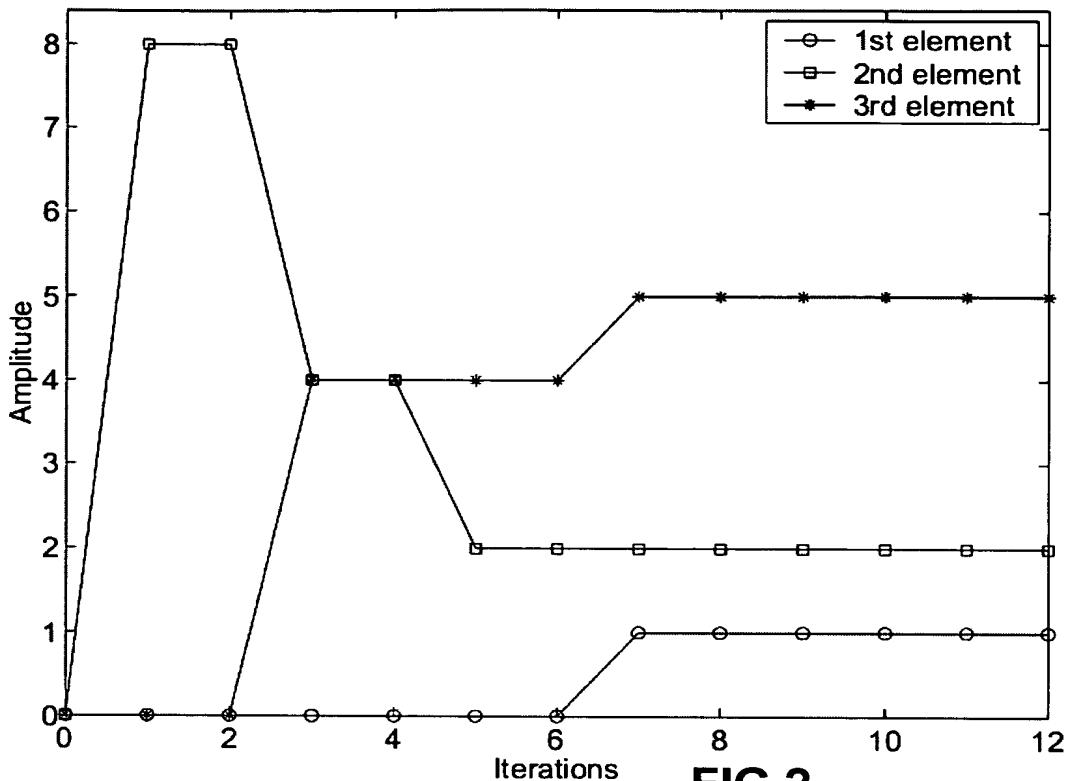
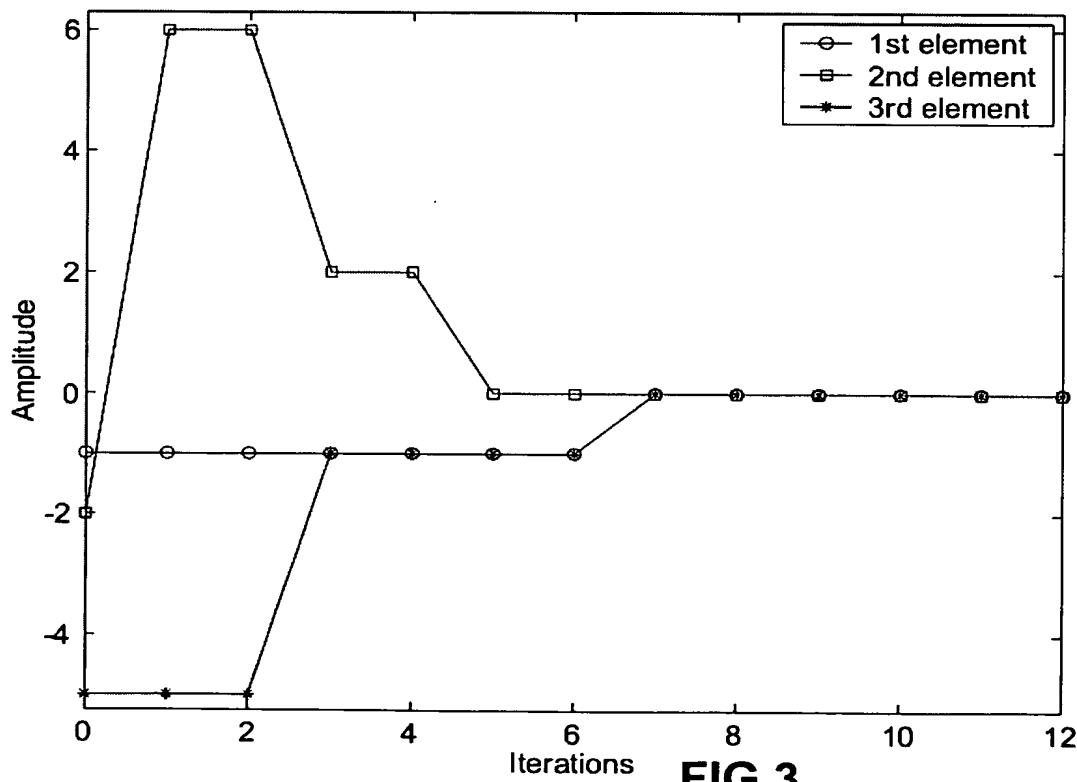
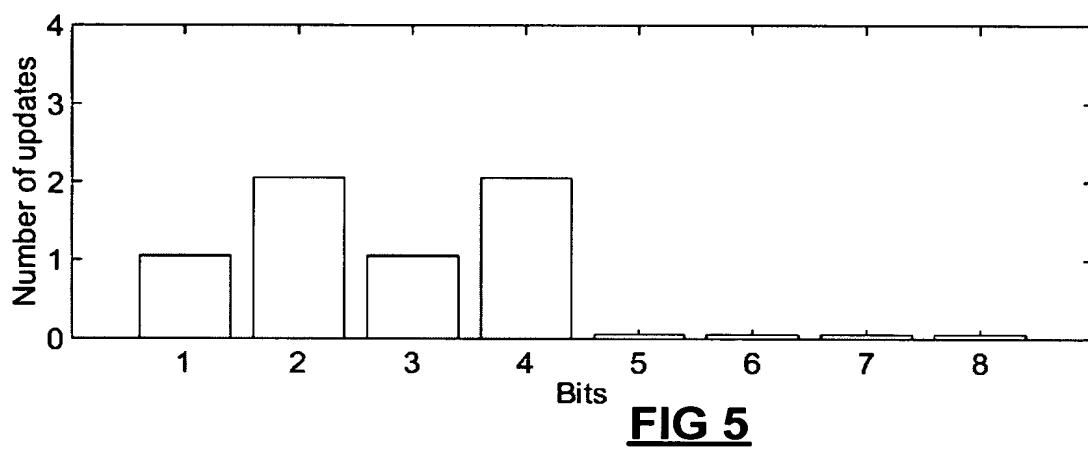
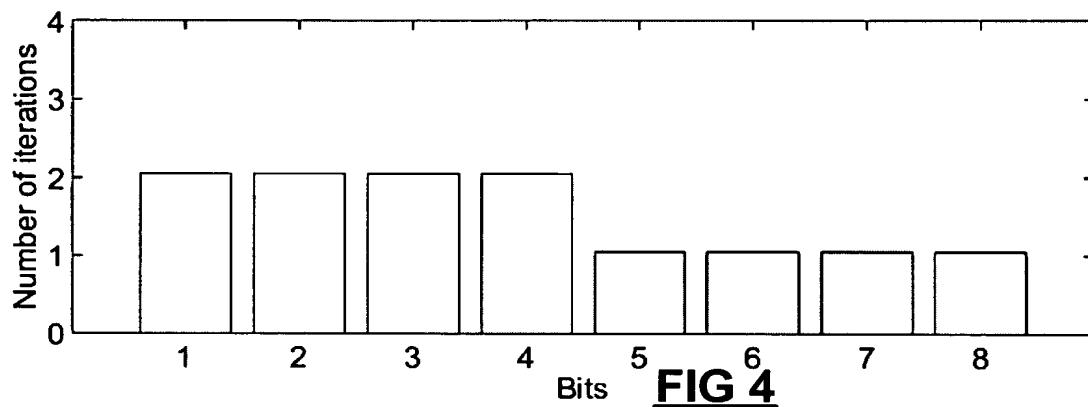
**FIG 1**

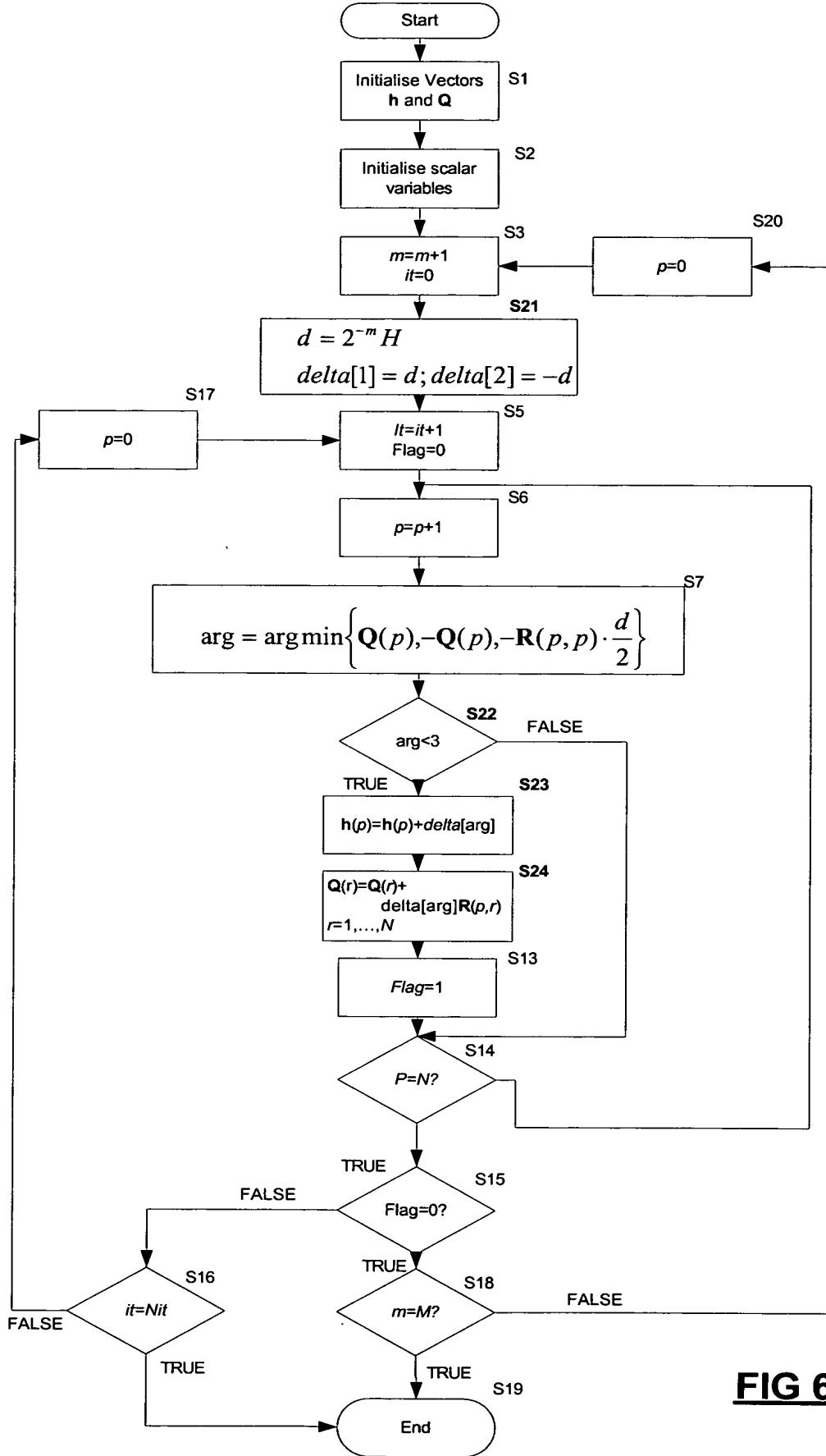
Solution vector vs. iterations

**FIG 2**

Error vs. iterations

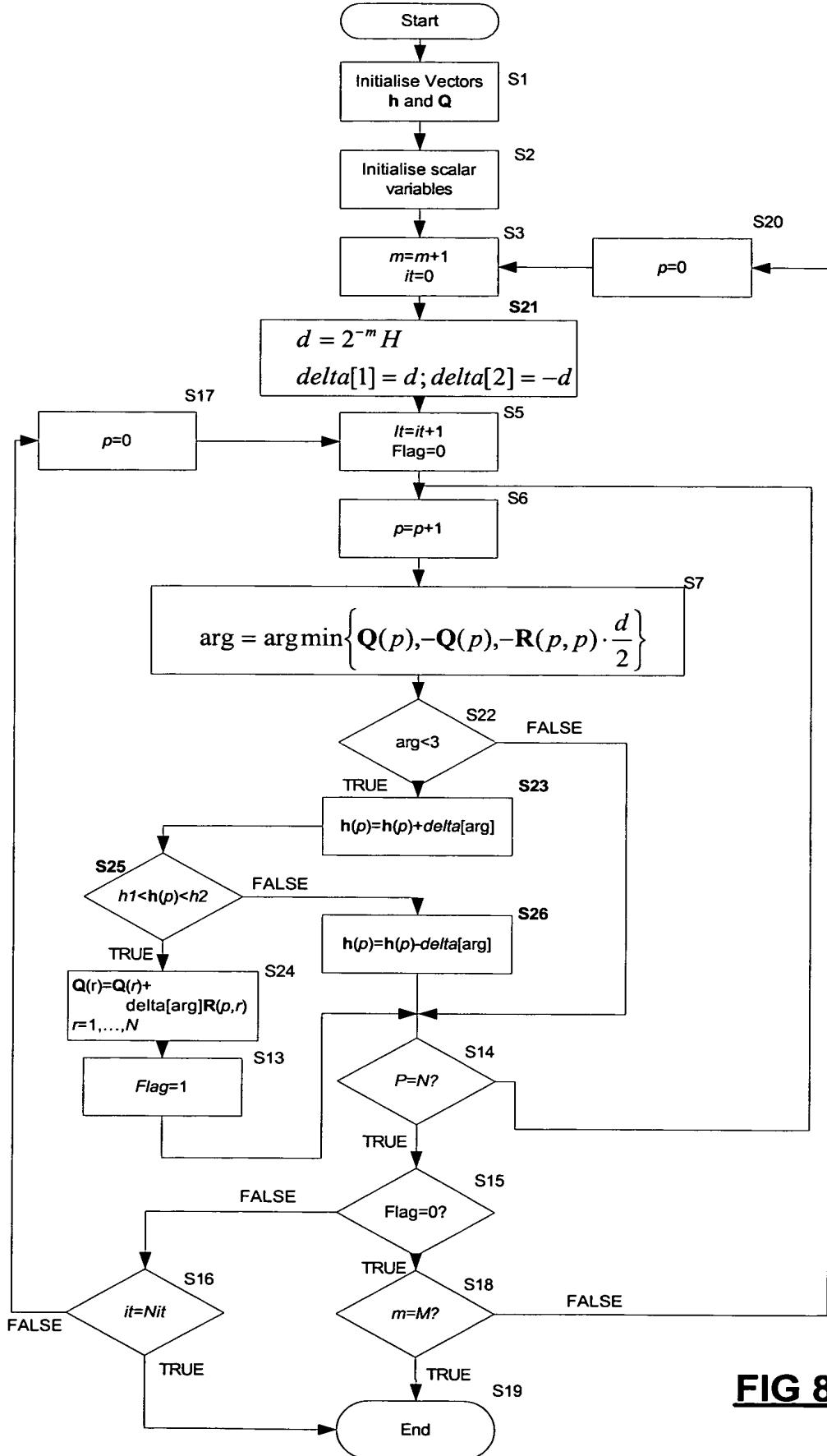
**FIG 3**



**FIG 6**

```
function h=LS_real(R,N,M,H,Nit,beta);
h=zeros(N,1);
delta=zeros(1,2);
Q=-beta;
d=H/2;
for m=1:M
    delta(1)=d; delta(2)=-d;
    d=d/2;
    for it=1:Nit
        Flag=0;
        for p=1:N
            [val arg]=min([Q(p), -Q(p), -R(p,p)*d]);
            if arg<3
                Flag=1;
                h(p)=h(p)+delta(arg);
                Q=Q+delta(arg)*R(p,:);
            end
        end
        if Flag==0 break; end
    end
end
return
```

FIG 7

**FIG 8**

$$\arg = \arg \min \left\{ \operatorname{Re}\{\mathbf{Q}(p)\}, -\operatorname{Re}\{\mathbf{Q}(p)\}, \operatorname{Im}\{\mathbf{Q}(p)\}, -\operatorname{Im}\{\mathbf{Q}(p)\}, -\mathbf{R}(p, p) \cdot \frac{d}{2} \right\}$$

S27

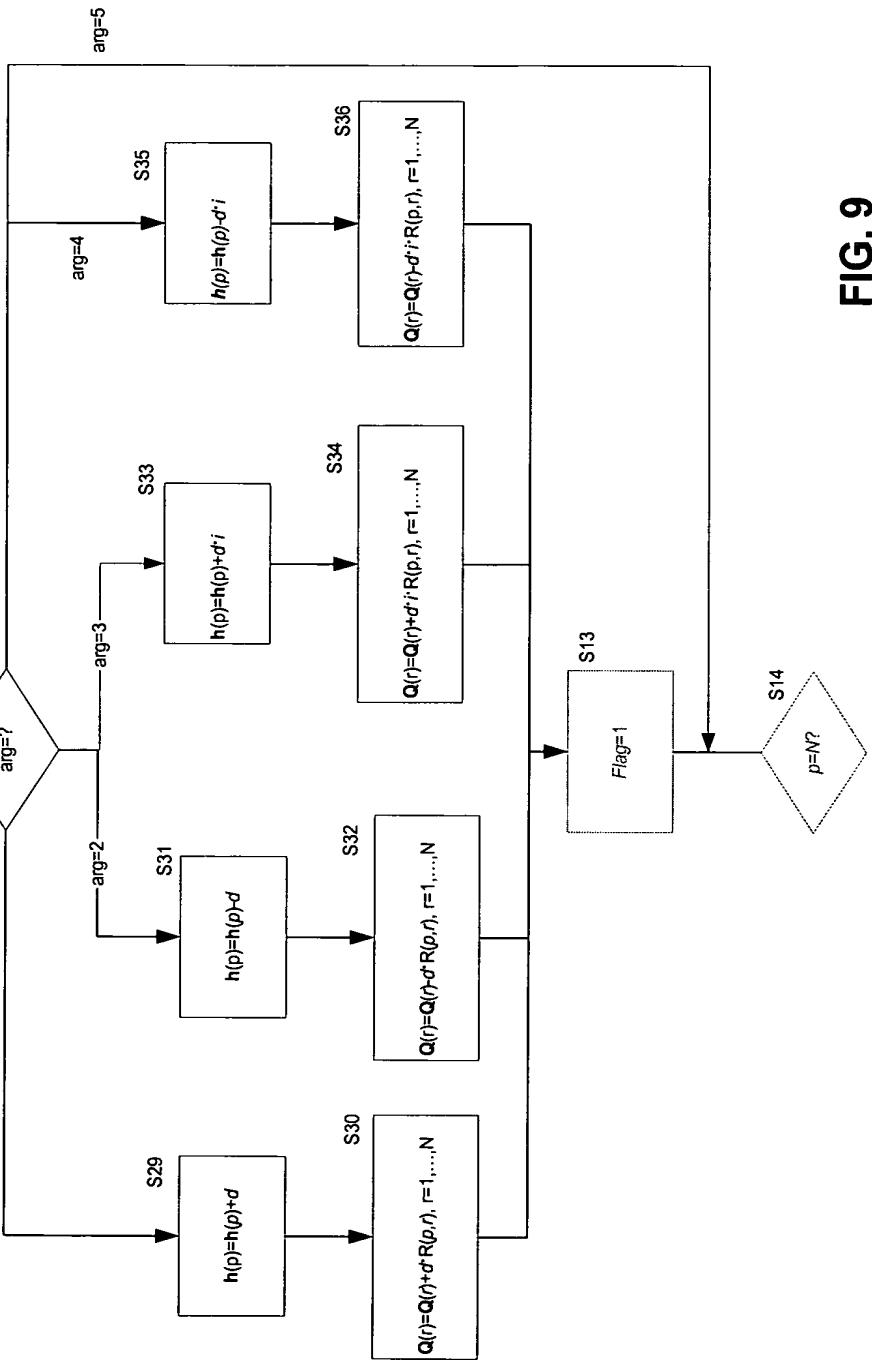
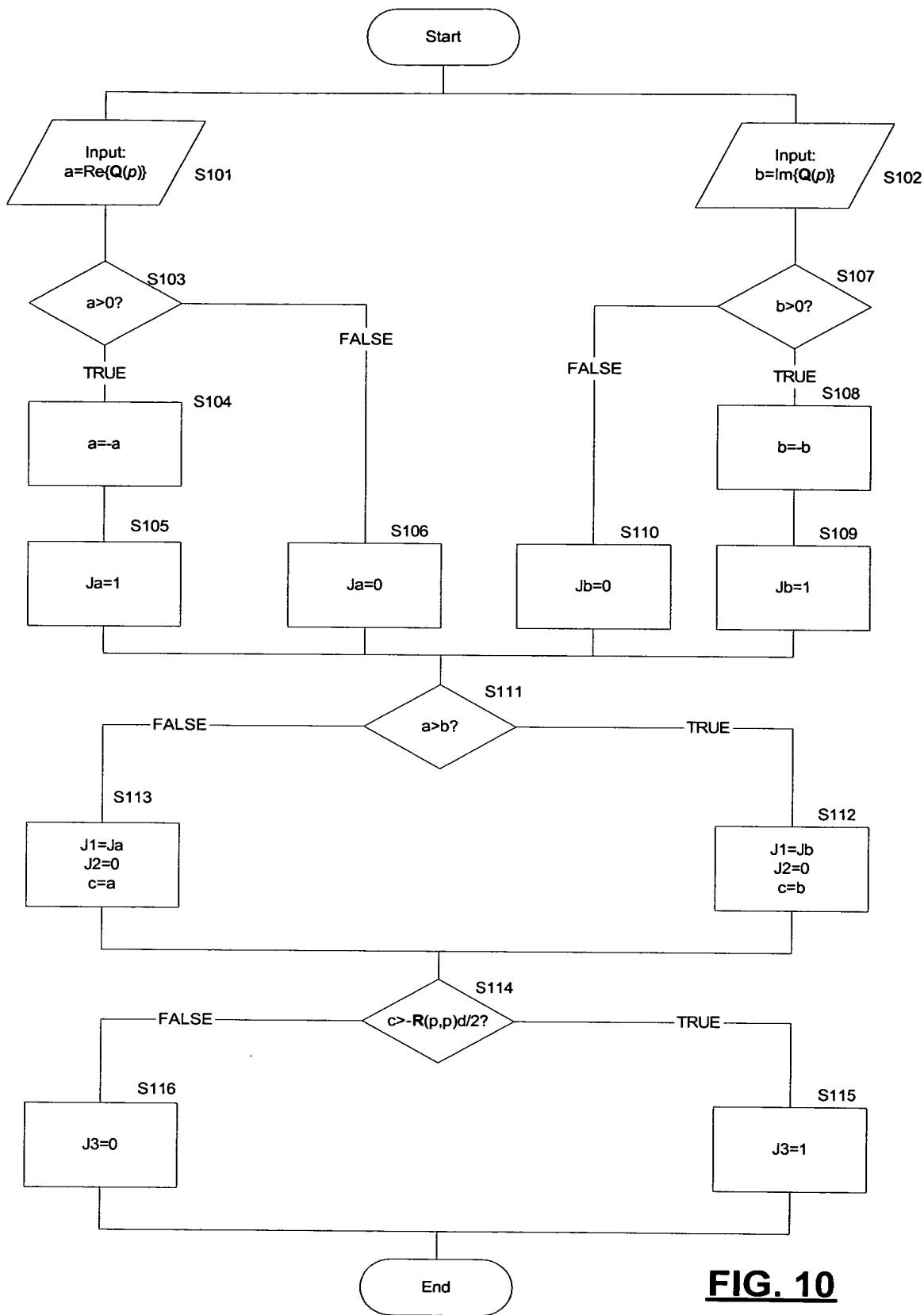


FIG. 9



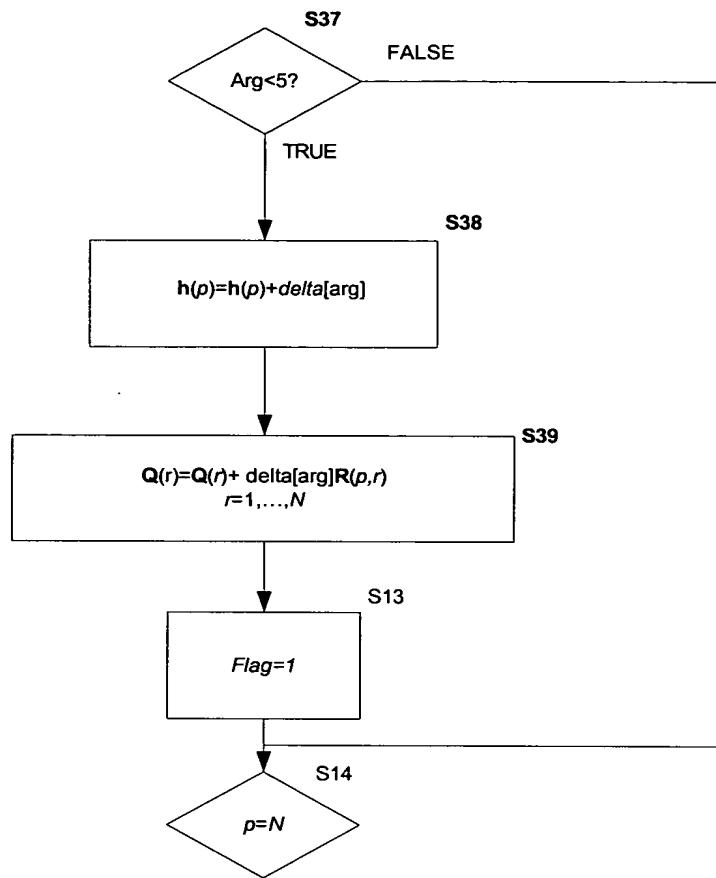


FIG 11

```
function h=LS_cmplx(R,N,M,H,Nit,beta);
h=zeros(N,1);
delta=zeros(1,4);
Q=-beta;
d=H/2;
for m=1:M
    delta(1)=d; delta(2)=-d; delta(3)=i*d; delta(4)=-i*d;
    d=d/2;
    for it=1:Nit
        Flag=0;
        for p=1:N
            [val arg]=min([real(Q(p)), -real(Q(p)), ...
                           imag(Q(p)), -imag(Q(p)), -R(p,p)*d]);
            if arg<5
                Flag=1;
                h(p)=h(p)+delta(arg);
                Q=Q+delta(arg)*R(p,:);
            end
        end
        if Flag==0 break; end
    end
end
return
```

FIG 12

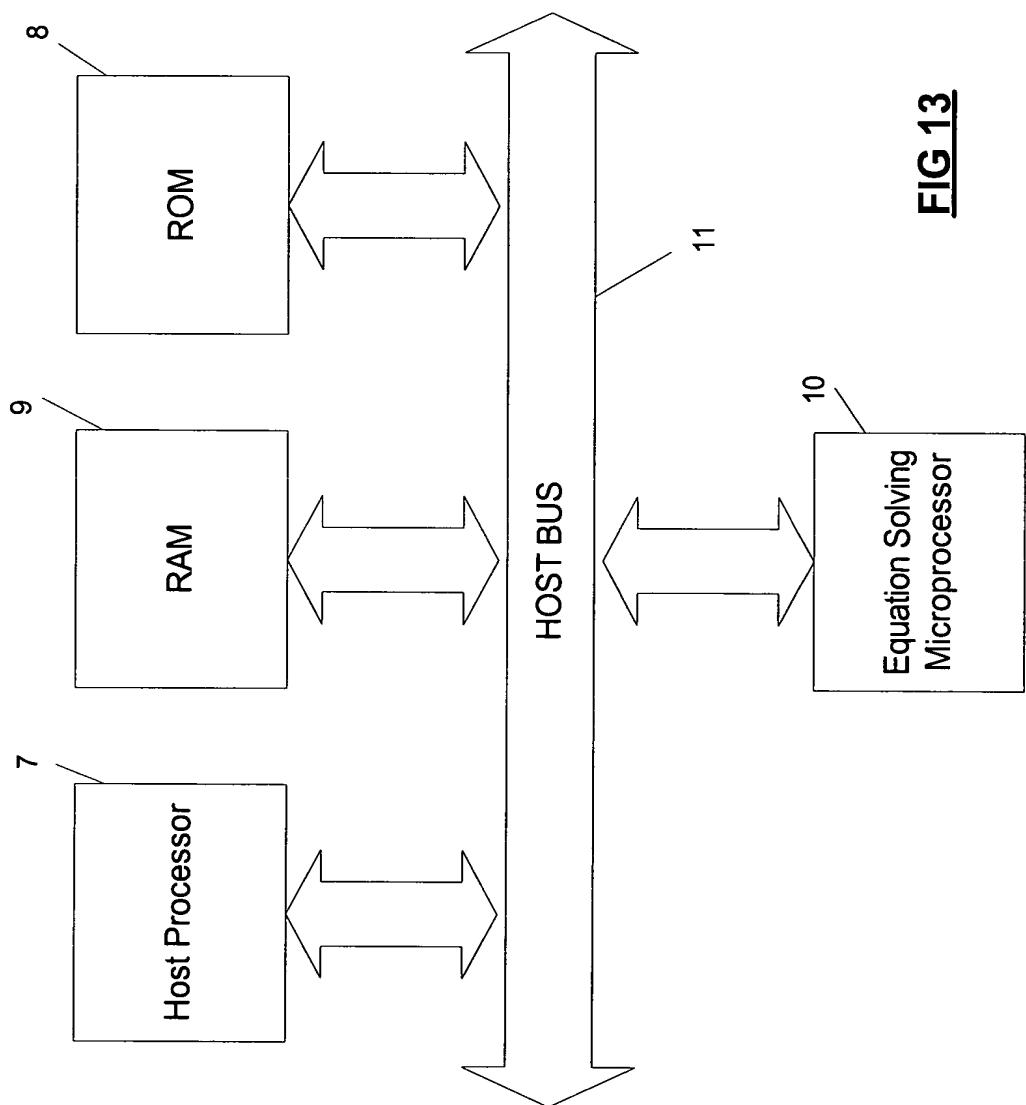


FIG 13

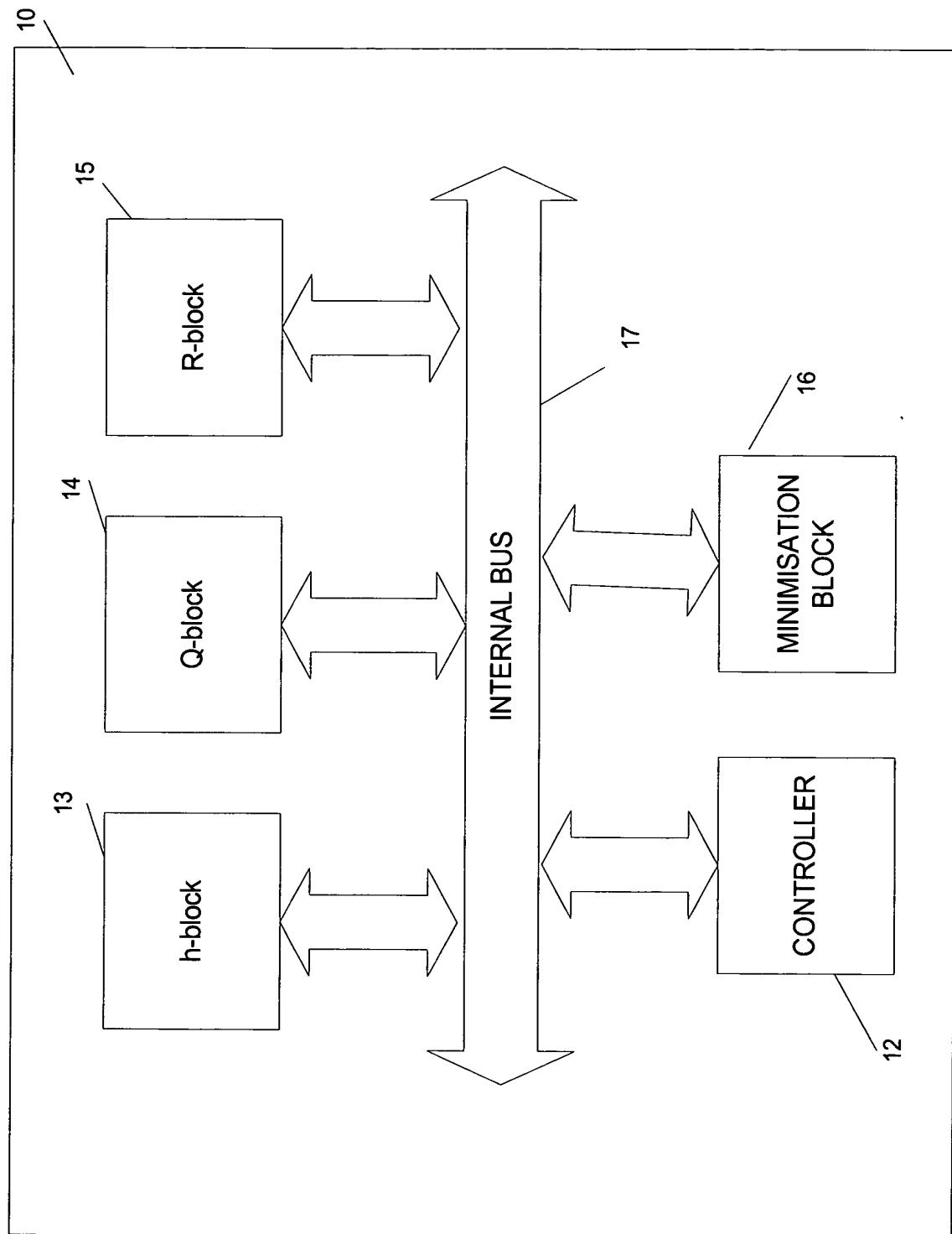
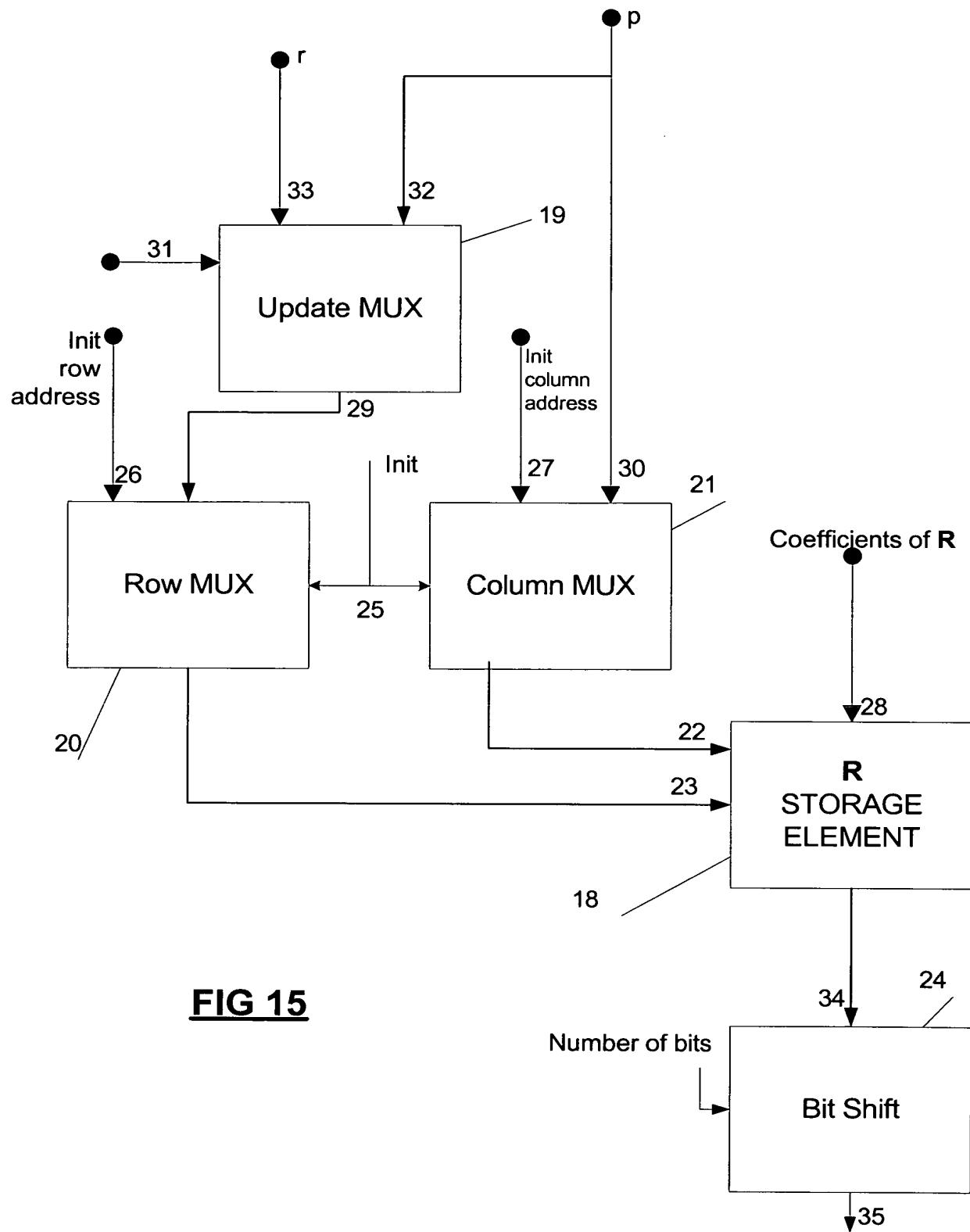
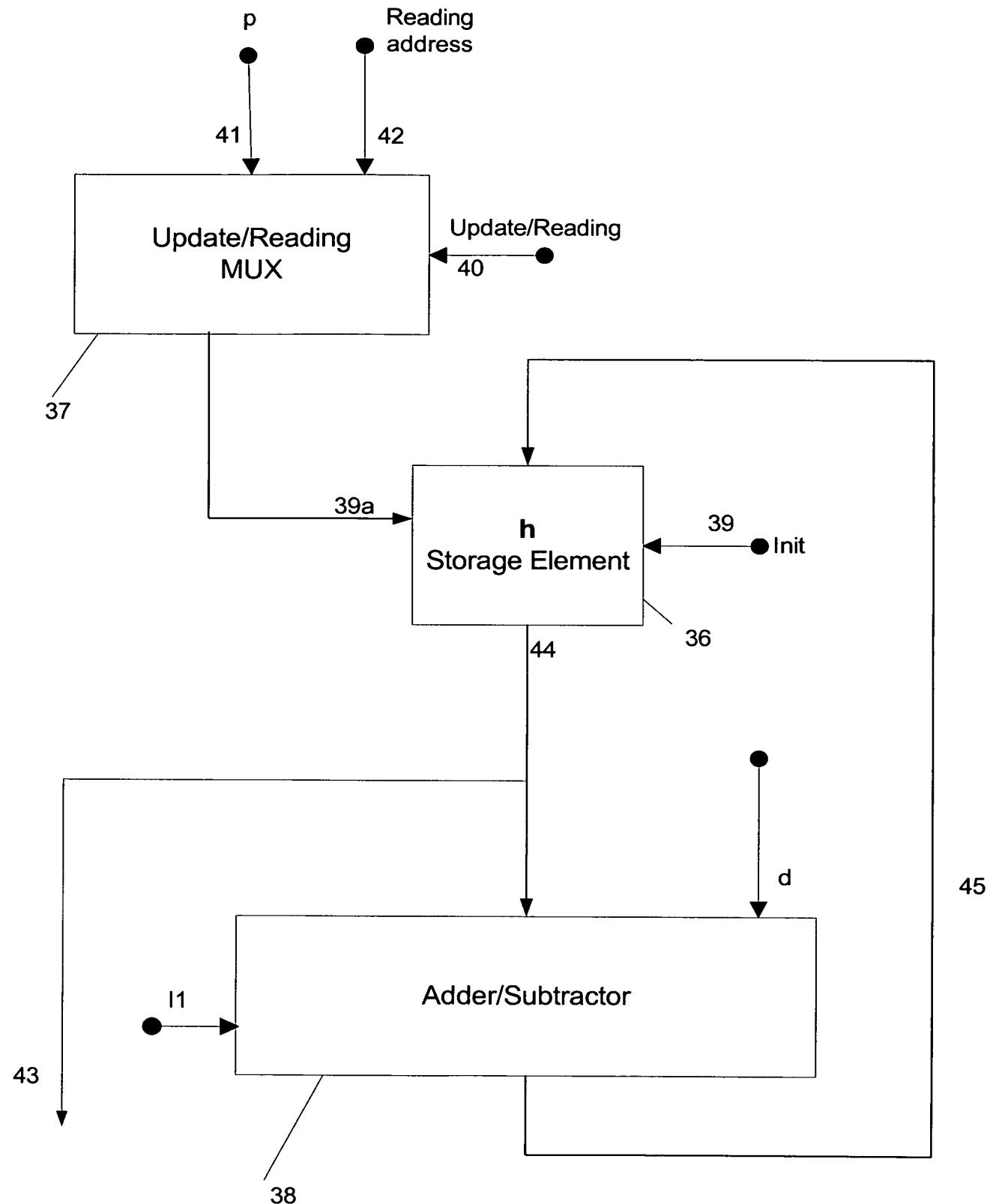
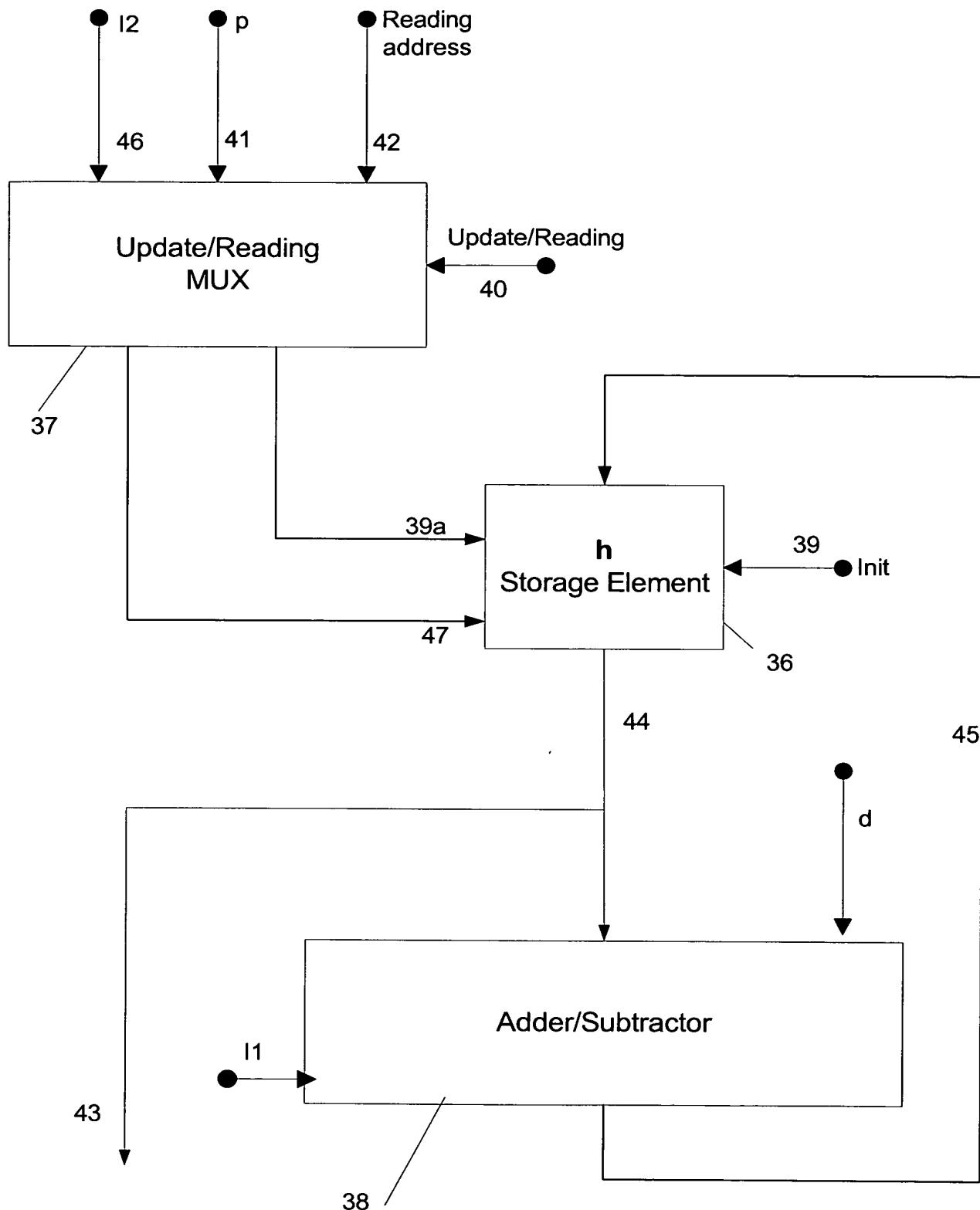
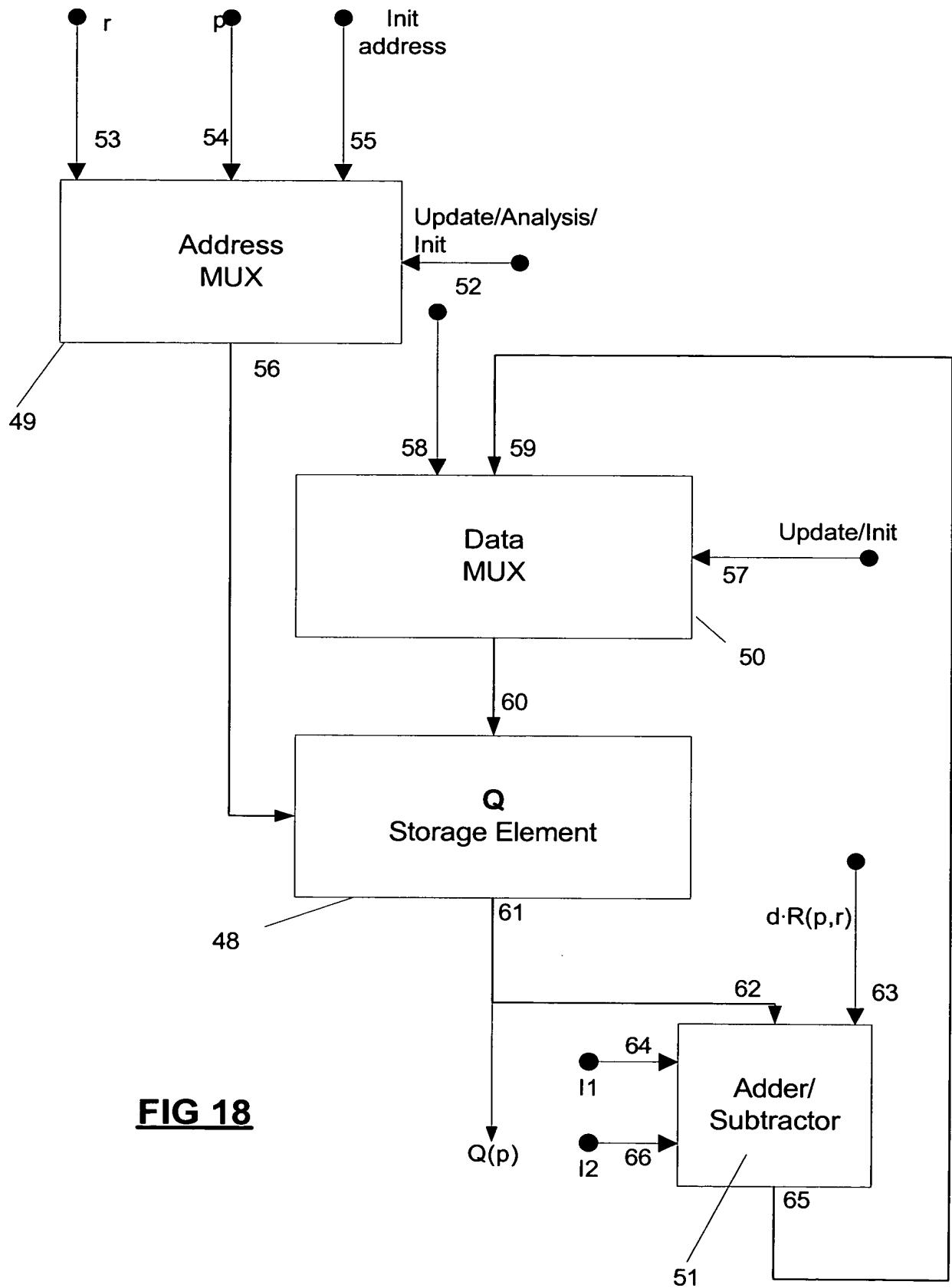


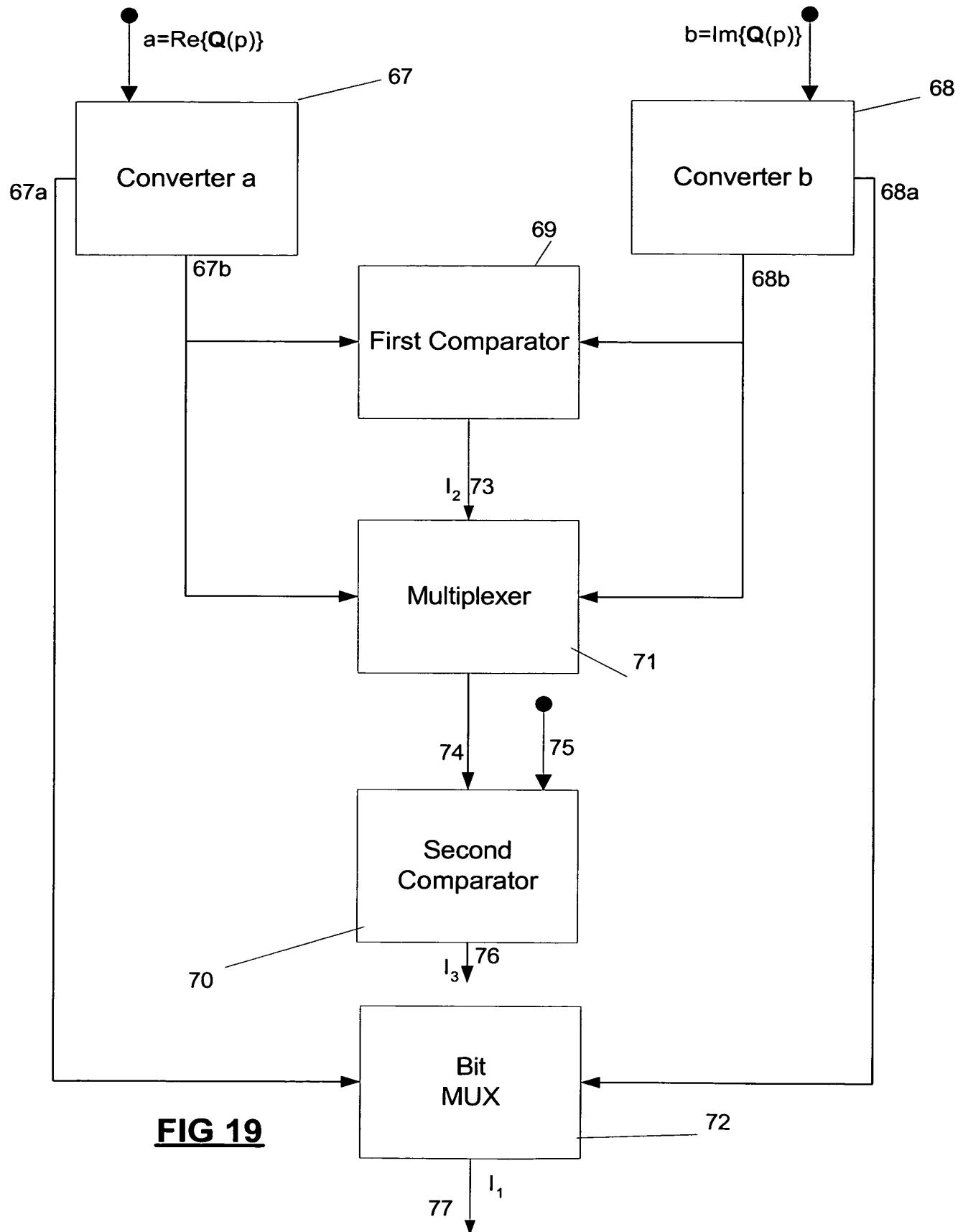
FIG 14

**FIG 15**

**FIG 16**

**FIG 17**



**FIG 19**

```
function h=LS_real_beta(R,N,M,H,Nit,beta);
h=zeros(N,1);
delta=zeros(1,2);
d=H/2;
for m=1:M
    delta(1)=d; delta(2)=-d;
    d=d/2;
    for it=1:Nit
        Flag=0;
        for p=1:N
            [val arg]=min([-beta(p),beta(p),-R(p,p)*d]);
            if arg<3
                Flag=1;
                h(p)=h(p)+delta(arg);
                beta=beta-delta(arg)*R(p,:);
            end
        end
        if Flag==0 break; end
    end
end
return
```

FIG 19a

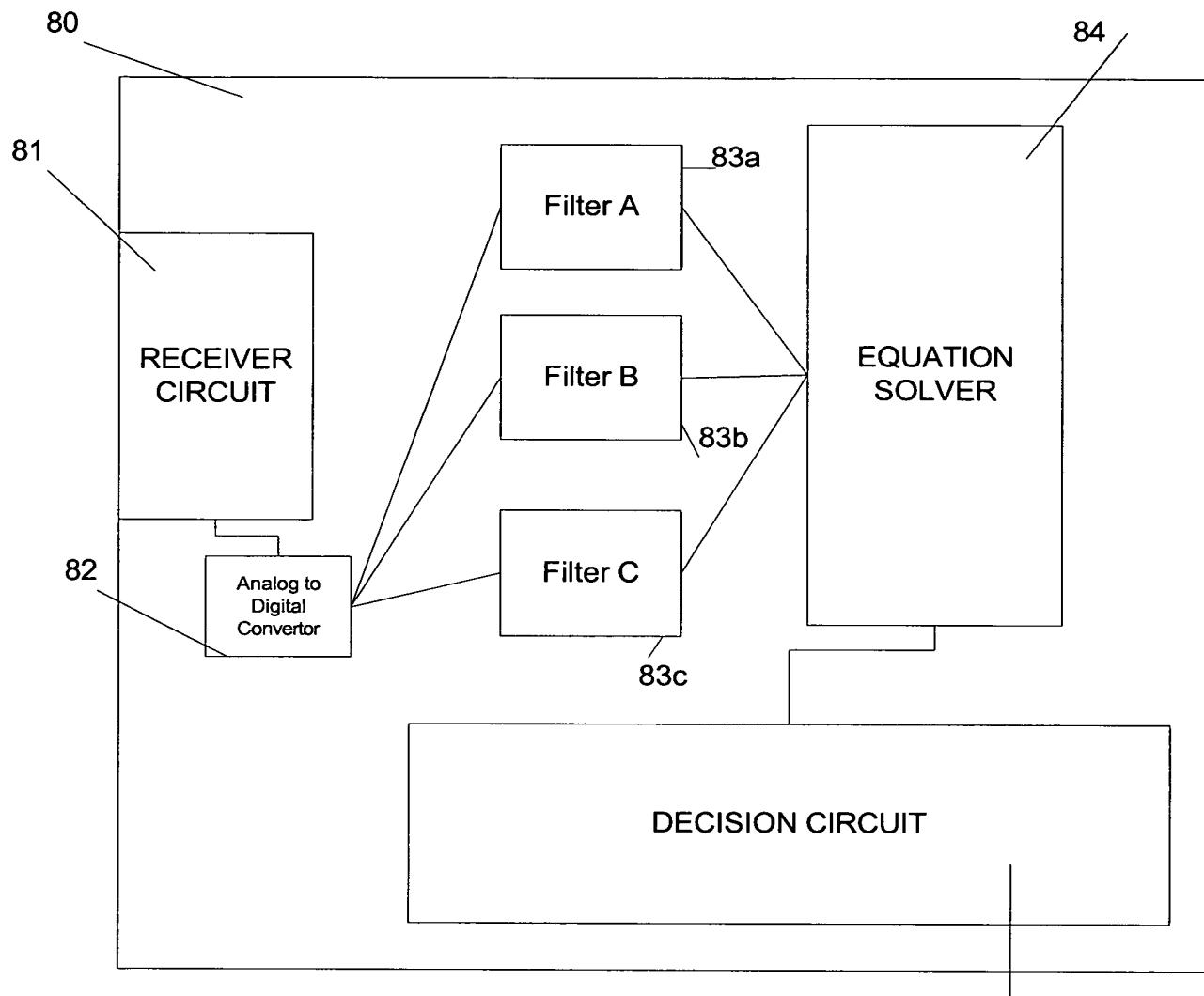


FIG 20

20/20

